

EVALUASI TOOLS UMLS BERBASIS OPEN SOURCE MENGUNAKAN FRAMEWORK DECIDE DENGAN PENDEKATAN DAN METODE GOMS

Harijanto Pangestu

Jurusan Sistem Informasi, Universitas Bina Nusantara,
Jl. KH. Syahdan 7-9, Kebon Jeruk, Jakarta 11480
Email : harijantopangestu@binus.ac.id

Abstrak

Unified Modeling Language (UML) adalah satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek (Whitten, 2004). Untuk mempermudah penggambaran UML telah dikembangkan perangkat lunak pemodelan visual baik berbasis open source maupun berbasis proprietary. Akan tetapi perangkat lunak ini tidak mudah digunakan dan banyak tools yang tersedia sehingga membingungkan pengguna. Tujuan penelitian ini adalah mengevaluasi tools UML yang berbasis visual dari sisi user interface dan usability. Tools yang diteliti dibatasi hanya perangkat lunak berbasis open source karena dapat diperoleh dengan mudah dan tidak dibutuhkan dana. Dengan adanya evaluasi ini diharapkan pengguna UML dapat menggunakan tools uml yang tepat sesuai dengan kebutuhan. Evaluasi ini menggunakan framework DECIDE yaitu sebuah kerangka kerja penuntun evaluasi yang terdiri dari: Determine the goals, Explore the questions, Choose the evaluation approach and methods, Identify the practical issues, Decide how to deal with the ethical issues, Evaluate, analyze, interpret and present the data. Pendekatan dan teknik evaluasi yang digunakan dalam evaluasi ini adalah pendekatan GOMS (Goals, Operators, Methods, Selection rules). Tools UML adalah: Umleditor, Umlpad, Violet UML editor, ArgoUML, DIA, StarUML, UMLet, Winbrello. Hasil akhir penelitian ini adalah rekomendasi tools UML open source yang memiliki usability yang baik berdasarkan kriteria-kriteria penilaian yang telah ditentukan. Setelah penelitian ini dilaksanakan maka ditarik kesimpulan bahwa UMLet 9.1 memiliki usability yang baik yaitu effective to use, safe to use, having good utility, easy to learn, easy to remember how to use walaupun untuk efficient to use mendapat posisi kedua dibandingkan dengan DIA.

Kata kunci: UML, Tools UML, Open Source, DECIDE

1. PENDAHULUAN

Unified Modeling Language (UML) adalah satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek (Whitten, 2004), UML membantu menspesifikasi, memvisualisasi, dan mendokumentasikan model-model dari sistem software, termasuk struktur dan desain, dengan mempertemukan keseluruhan spesifikasi ini dengan kebutuhan-kebutuhan sistem.

Untuk mempermudah penggambaran UML telah dikembangkan perangkat lunak pemodelan visual baik berbasis open source maupun berbasis proprietary (berbayar). Akan tetapi perangkat lunak pemodelan visual tidak mudah digunakan dan banyak tools yang tersedia sehingga membingungkan pengguna.

Perangkat lunak pemodelan visual yang diteliti dibatasi hanya perangkat lunak pemodelan visual berbasis open source karena perangkat lunak tersebut mudah diperoleh dan untuk mendapatkannya tidak dibutuhkan dana. Tools UML opensource yang akan dievaluasi adalah: Umleditor versi 3.0.1, Umlpad versi 3.2, Violet UML editor versi 0.21.1, ArgoUML versi 0.26.2, Dia versi 0.97, StarUML versi 5.0.2.1570, UMLet versi 9.1, Winbrello (Umbrello UML Modeller versi 1.5.71)

Perangkat lunak pemodelan visual ini dievaluasi dari sisi user interface dan usability sehingga pengguna UML dapat menggunakan tools uml yang tepat sesuai dengan kebutuhan. Tools UML open source yang dievaluasi diharapkan dapat membantu pengembangan sistem mulai dari tahap analisa, desain, sampai dengan tahap implementasi. Karena itu diagram UML yang diuji coba meliputi *Use-Case Model Diagrams, Class diagrams, Sequence diagrams, Activity diagrams, Component diagrams* serta *Deployment diagrams*. Tujuan penelitian ini adalah mengevaluasi perangkat lunak pemodelan visual berbasis open source sebagai tools UML dari sisi user interface dan usability.

2. TINJAUAN PUSTAKA

Menurut Borgx, Clouseth, HarisX, IvanLanin, Kemal Prihatman, Relly Komaruzaman (2010) Sumber terbuka (Inggris: open source) adalah sistem pengembangan yang tidak dikoordinasi oleh suatu individu / lembaga pusat, tetapi oleh para pelaku yang bekerja sama dengan memanfaatkan kode sumber (source-code) yang tersebar dan tersedia bebas (biasanya menggunakan fasilitas komunikasi internet). Pola pengembangan ini mengambil model ala bazaar, sehingga pola Open Source ini memiliki ciri bagi komunitasnya yaitu adanya

dorongan yang bersumber dari budaya memberi, yang artinya ketika suatu komunitas menggunakan sebuah program Open Source dan telah menerima sebuah manfaat kemudian akan termotivasi untuk menimbulkan sebuah pertanyaan apa yang bisa pengguna berikan balik kepada orang banyak.

Menurut Helen Sharp, Yvonne Rogers dan Jenny Preece (2006) Kerangka Kerja DECIDE merupakan sebuah kerangka kerja penuntun evaluasi. Kerangka kerja ini terdiri dari:

- Tentukan Sasaran (*Determine the goals*).
Menentukan tujuan evaluasi adalah hal pertama yang harus dilakukan. Apa tujuan evaluasi? Siapa yang menginginkan evaluasi? Mengapa harus melakukan evaluasi? Tujuan akan menentukan ruang lingkup evaluasi sehingga mengidentifikasi tujuan merupakan langkah pertama dalam perencanaan evaluasi.
- Eksplorasi Pertanyaan (*Explore the questions*).
Untuk mencapai tujuan yang telah ditentukan di atas maka dibuat pertanyaan-pertanyaan yang harus dijawab sebagai bahan-bahan yang akan dipelajari dalam evaluasi.
- Memilih Pendekatan dan Metode Evaluasi (*Choose the evaluation approach and methods*).
Menurut Helen Sharp, Yvonne Rogers dan Jenny Preece (2006) ada tiga pendekatan evaluasi yaitu :
 - (1) Usability testing
Usability testing melibatkan tipe-tipe pengguna tertentu pada tipe tugas tertentu. Jumlah dan macam-macam kesalahan yang dibuat pengguna dicatat demikian pula dengan waktu yang dibutuhkan pengguna untuk menyelesaikan tugas tersebut. Kegiatan pengguna diawasi dan direkam dengan menggunakan video recorder.
 - (2) Field studies
Pendekatan kedua adalah studi lapangan. Pengguna melakukan kegiatan dalam lingkungan yang sewajarnya. Pendekatan ini bertujuan untuk memahami bagaimana pengguna melakukan kegiatan dan bagaimana produk dapat membantu kegiatan pengguna.
 - (3) Analytical evaluation
Pendekatan ini tidak melibatkan pengguna. Evaluasi dilakukan oleh ahli yang berhubungan dengan produk tersebut.
- Identifikasi Isu-Isu Praktis (*Identify the practical issues*).
Isu-Isu praktis yang antara lain perlu diidentifikasi adalah bagaimana memilih *user*, bagaimana mentaati anggaran dan jadwal, bagaimana mencari evaluator, bagaimana memilih peralatan.
Mengenali tingkat keahlian pemakai menurut Ben Shneiderman dan Catherine Plaisant (2005) :
 - (1) Novice (first-time users)
Novice users adalah profesional yang mengerti konsep tentang tugasnya tetapi memiliki konsep pengetahuan antarmuka yang dangkal. Perancangan yang cocok adalah dengan membatasi jumlah pilihan, umpan balik yang informatif, user manual dan tutorial online yang efektif.
 - (2) Knowledgeable intermittent users
Adalah profesional yang telah memiliki konsep tugas yang stabil dan konsep antarmuka yang luas namun sulit mengingat atau menguasai struktur menu atau lokasi-lokasi fitur yang tersedia. Perancangan yang cocok adalah dengan menyediakan struktur menu yang rapi, konsisten, kejelasan antarmuka, perlindungan dari bahaya karena eksplorasi fitur.
 - (3) Expert frequent users
Adalah profesional yang terbiasa dengan konsep tugas dan antarmuka. Mereka ingin menyelesaikan pekerjaannya dengan cepat. Perancangan yang cocok adalah dengan menyediakan makro, *shortcuts*, singkatan dan lain sebagainya.
- Tentukan Isu-Isu Etis (*Decide how to deal with the ethical issues*).
Dalam melakukan evaluasi harus memperhatikan masalah-masalah yang berkaitan dengan etika. Data pribadi dan hal-hal yang menunjuk kepada identitas seseorang harus dijaga kerahasiaannya. Partisipan mempunyai hak untuk mengetahui sasaran evaluasi, apa yang akan dilakukan terhadap hasil evaluasi, mendapat perlindungan (privasi) atas informasi pribadi, tidak dikutip tanpa persetujuan, meninggalkan tempat kapan pun mereka suka, diperlakukan secara sopan.
- Evaluasi, Analisis, Interpretasi dan Presentasi Data (*Evaluate, analyze, interpret and present the data*).
Data yang diperoleh saat melakukan evaluasi harus dievaluasi dan dianalisa supaya data yang akan disajikan benar-benar valid sehingga pada saat data diinterpretasikan dan disajikan tidak salah. Bagaimana data dianalisa dan dipresentasikan tergantung pada paradigma dan teknik yang digunakan.

GOMS

GOMS merupakan singkatan dari *Goals, Operators, Methods, dan Selection rules*. GOMS merupakan salah satu metode untuk menganalisa suatu tugas (*task*) yang dikembangkan oleh Stuart Card, Thomas P. Moran and Allen Newell pada tahun 1983 di dalam bukunya *The Psychology of Human Computer Interaction*.

GOMS singkatan dari *Goals, Operators, Methods, dan Selection Rules*. *Goals* berarti tujuan yang ingin dicapai oleh pengguna, *Operators* berarti tindakan dasar yang harus dilakukan pengguna di dalam menggunakan

sistem, *Methods* berarti cara-cara yang dapat dilakukan untuk mencapai tujuan dengan menggunakan *operators* yang ada, dan *Selection Rules* berarti pilihan metode yang digunakan.

Metode GOMS tidak memberikan suatu hasil perhitungan yang akurat tentang bagaimana pengguna berinteraksi dengan sistem tetapi metode GOMS memberikan estimasi sebagai prediksi waktu yang dibutuhkan untuk mengerjakan suatu tugas dalam berhubungan dengan sistem.

Dalam melakukan analisa suatu tujuan dapat didekomposisi menjadi tujuan-tujuan yang lebih kecil yang selanjutnya sampai dengan level operator dasar. Dengan demikian maka GOMS merupakan suatu metode yang baik untuk mendeskripsikan bagaimana seorang ahli melakukan suatu tugas dalam hubungannya dengan suatu sistem yang sedang diuji. Karena itu GOMS bukan untuk menganalisis tugas-tugas dimana langkah-langkahnya kurang dipahami dan tidak dilakukan pada pengguna awam atau yang tidak berpengalaman.

Keystroke Level Model (KLM)

KLM adalah suatu metode untuk menganalisa suatu tugas (*task*) yang dikembangkan oleh Stuart Card, Thomas P. Moran and Allen Newell pada tahun 1983 di dalam bukunya *The Psychology of Human Computer Interaction*. KLM merupakan bentuk model GOMS tingkat terendah.

Ketentuan *operators* dan waktu pengerjaan menurut Helen Sharp, Yvonne Rogers dan Jenny Preece (2006) :

Tabel 1. Ketentuan operator dan waktu pengerjaan

Operator Name	Description	Time
K	Pressing a single key or button	0.35 (average)
	Skilled typist (55 wpm)	0.22
	Average typist (40 wpm)	0.28
	User unfamiliar with the keyboard	1.20
	Pressing shift or control key	0.08
P	Pointing with a mouse or other device to a target on a display	1.10
P _c	Clicking the mouse or similar device	0.20
H	Homing hands on the keyboard or other device	0.40
D	Draw a line using a mouse	Variable depending on the length of line
M	Mentally prepare to do something (e.g. make a decision)	1.35
R (t)	System response time – counted only if it causes the user to wait when carrying out their task	t

Operator "K - Keystroking" berarti menekan suatu tuts pada papan tuts, operator "P - Pointing" berarti memindahkan mouse ke target atau menekan tombol pada mouse, operator "H - Homing" berarti memindahkan tangan dari mouse ke papan tuts, operator "D - Drawing" berarti menggambar dengan menggunakan mouse, operator "M - Mental Preparation" berarti mempersiapkan sikap mental untuk suatu aksi fisik, operator "R - Response" berarti respons sistem dalam waktu ketika pengguna harus menunggu dalam usaha menyelesaikan suatu tugas.

Eight golden rules of interface design (delapan aturan emas perancangan user interface) menurut Ben Shneiderman dan Catherine Plaisant (2005) :

1. *Strive for consistency* (berusaha untuk konsisten).
2. *Cater to universal usability* (memungkinkan *frequent users* menggunakan *shortcuts*).
3. *Offer informative feedback* (memberikan umpan balik yang informatif).
4. *Design dialogs to yield closure* (merancang dialog yang memberikan penutupan (keadaan akhir)).
5. *Prevent errors* (memberikan pencegahan kesalahan dan penanganan kesalahan yang sederhana).
6. *Permit easy reversal of actions* (memungkinkan pembalikan aksi yang mudah).
7. *Support internal locus of control* (mendukung pusat kendali internal).
8. *Reduce short-term memory load* (mengurangi beban ingatan jangka pendek).

Menurut Helen Sharp, Yvonne Rogers dan Jenny Preece (2006) *Usability* pada umumnya adalah mengenai tingkat kepastian dimana sebuah produk mudah dipelajari, efektif, dan menyenangkan menggunakannya dari sudut pandang pengguna. Secara khusus, tujuan *usability* adalah:

- Effective to use (effectiveness)
- Efficient to use (efficiency)
- Safe to use (safety)
- Having good utility (utility)
- Easy to learn (learnability)
- Easy to remember how to use (memorability)

Usability merupakan hal yang sangat penting karena jika sebuah produk memiliki fungsi yang baik tetapi tidak memiliki *usability* yang baik maka produk tersebut tidak dapat digunakan dengan maksimal dan kemungkinan produk tersebut tidak dapat dipakai dan ditinggalkan oleh penggunanya. *Usability* yang tinggi akan menjamin sebuah produk dapat digunakan dengan efektif, efisien, aman digunakan, memiliki fungsi yang baik, mudah dipelajari dan mudah digunakan.

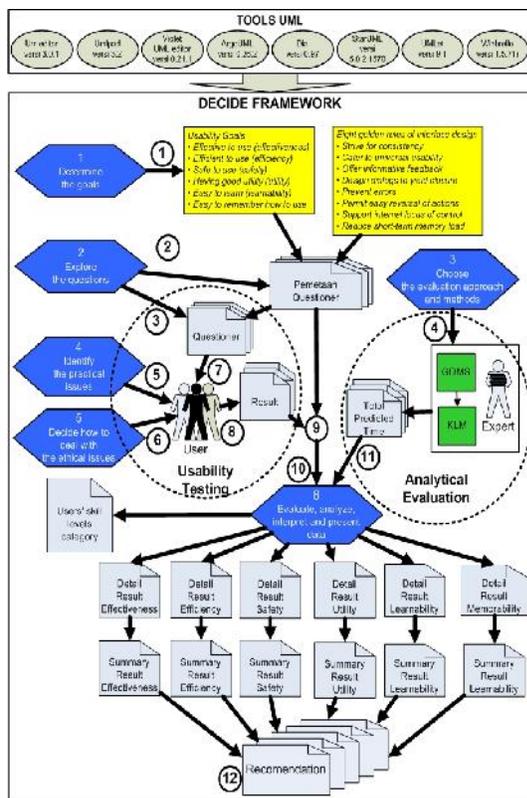
Penelitian sebelumnya bersumber dari *Master Thesis Computer Science, Thesis no: MCS-2009-10*, dengan judul "*Usability assessment Method of the open source applications (Case Study of OpenOffice.Org 3.0)*" yang ditulis oleh Syed Mehr Ali Shah dari School of Computing Blekinge Institute of Technology Soft Center SE-37225 Ronneby Sweden. *OpenOffice.org* dapat digunakan pada sistem operasi yang berbeda-beda dan dengan bahasa yang berbeda-beda. Fitur *OpenOffice.org 3.0* mirip dengan fitur Microsoft Office 2003. *OpenOffice.org 3.0* memiliki aplikasi *word processing*, *spreadsheet* dan aplikasi *presentation* yang ada bersama-sama dalam satu kesatuan. Tetapi aplikasi ini tidak begitu populer untuk pengguna umum yang menyebabkan hambatan besar dalam mempelajari *usability*. Tujuan utama dari penelitian pertama ini adalah untuk menemukan

isu-isu dan *users satisfaction* yang berhubungan dengan OpenOffice.org. Metode empiris untuk menilai *usability* yang digunakan adalah *thinks aloud*, *Questionnaires* dan *interviews*. Perbedaan dengan penelitian ini adalah objek penelitian sebelumnya adalah *OpenOffice.org 3.0* sedangkan penelitian ini adalah *tools uml* berbasis *open source*. Penelitian sebelumnya hanya menggunakan *usability testing* tanpa konfirmasi *analytical evaluation*. Dalam penelitian sebelumnya metode *usability testing* yang digunakan adalah *thinks aloud*, *Questionnaires* dan *interviews* sedangkan dalam penelitian ini *thinks aloud* dan *interviews* tidak digunakan.

Penelitian kedua sebelumnya berjudul "*2 Usability Evaluation of User Interfaces Generated with a Model-Driven Architecture Tool*" yang ditulis oleh Silvia Abrahão, Emilio Iborra, Jean Vanderdonckt meneliti *CASE Tool* yang digunakan dalam bidang Software Engineering. Perangkat lunak yang dievaluasi dalam penelitian ini adalah perangkat lunak *OlivaNova*. Tujuan dari *experimental study* adalah untuk mengevaluasi tingkat *usability* dari antarmuka pengguna yang telah dibuat secara otomatis menggunakan sebuah *MDA tool*. Untuk melaksanakan hal tersebut dipilih dua evaluasi *usability* yaitu *Action Analysis* (Olson and Olson 1990) dan *user testing* (Dumas and Redish, 1999). Dua metode evaluasi *usability* dipilih karena dapat saling melengkapi. Perbedaan dengan penelitian ini adalah objek penelitian yang digunakan dalam penelitian sebelumnya hanya satu saja yaitu perangkat lunak *OlivaNova* sedangkan penelitian ini menggunakan delapan tools UML open source. Penelitian sebelumnya bertujuan mengevaluasi tingkat *usability* dari antarmuka pengguna yang telah dibuat secara otomatis menggunakan sebuah *MDA tool* sedangkan dalam penelitian ini tujuannya adalah mengevaluasi perangkat lunak pemodelan visual berbasis open source sebagai tools UML dari sisi user interface dan *usability*.

3. METODE PENELITIAN

Gambar 1. Penggunaan Kerangka Kerja DECIDE dan Metode GOMS – KLM dalam mengevaluasi Tools UML berbasis Open Source.



- Keterangan (gambar 1) :
- (1) Menentukan tujuan evaluasi.
 - (2) Eksplorasi Pertanyaan dengan membuat pemetaan antara tujuan evaluasi dengan Questioner yang akan dibuat.
 - (3) Membuat Questioner yang sesuai dengan tujuan evaluasi dan berpedoman pada Usability Goals dan eight golden rules of interface design.
 - (4) Memilih Metode Evaluasi dengan menggunakan GOMS dan KLM sehingga menghasilkan Total Predicted Time sebagai hasil dari Analytical Evaluation.
 - (5) Memilih Pengguna sesuai tingkat keahlian pengguna.
 - (6) Dalam melakukan evaluasi harus memperhatikan masalah-masalah yang berkaitan dengan etika.
 - (7) Questioner diberikan kepada pengguna sebagai pelaksanaan Usability testing.
 - (8) Hasil pengisian Questioner merupakan hasil dari Usability testing.
 - (9) Analisis dan klasifikasi hasil pengisian Questioner berdasarkan Usability Goals dan eight golden rules of interface design.
 - (10) Hasil Analisis dan klasifikasi pengisian Questioner digunakan untuk membandingkan antara hasil Usability Testing dengan hasil Analytical Evaluation.
 - (11) Total Predicted Time sebagai hasil dari Analytical Evaluation.
 - (12) Memberikan suatu rekomendasi sebagai hasil analisis Usability Testing dengan Analytical Evaluation.

Contoh GOMS-UseCase

Goal: Menggambar simbol UseCase

Operators: Move-mouse
 Click mouse button
 Type Characters

Methods:

Untuk menggambar simbol UseCase

1. Click→move-mouse→click (*click methods*)

Untuk memberi nama UseCase

1. Type Characters (*type characters methods*)

Selection rules:

Untuk menggambar simbol UseCase *click methods*.

Untuk memberi nama UseCase gunakan *type characters methods*.

Keystroke Level Model (KLM)

Keystroke Level Model merupakan suatu metode untuk menganalisa suatu tugas (*task*). KLM merupakan bentuk model GOMS tingkat terendah. Hasil KLM dari tugas *Use Case Diagram* yang dikerjakan oleh penulis akan digunakan untuk memilih tools UML dalam hal mengerjakan tugas-tugas diagram UML berikutnya. Tiga Tools UML yang terpilih akan digunakan dalam kuestioner. Hasil akhir dari KLM untuk seluruh tugas menggambar diagram akan dibandingkan dengan hasil kuestioner yang dikerjakan oleh responden.

Tabel 2. Contoh Keystroke Level Model (KLM) untuk Menggambar Use Case "Withdraw cash" menggunakan click methods :

Description	Operator	Duration (sec)	Description	Operator	Duration (sec)
Mentally prepare	M	1.35	Type "r"	K	0.22
Move cursor to icon "UseCase" in toolbox	P	1.10	Type "h"	K	0.22
Click icon "UseCase"	P ₁	0.20	Type "d"	K	0.22
Mentally prepare	M	1.35	Type "r"	K	0.22
Move cursor to stack	P	1.10	Type "a"	K	0.22
Click cursor to draw	P ₁	0.20	Type "w"	K	0.22
Homeing hands on the keyboard	H	0.40	Type "space"	K	0.22
Mentally prepare	M	1.35	Type "C"	K	0.22
Pressing shift key	K	0.35	Type "a"	K	0.22
Type "W" (good typist)	K	0.22	Type "s"	K	0.22
Type "I"	K	0.22	Type "h"	K	0.22
Total predicted time			10.26		

4. HASIL DAN PEMBAHASAN

Diagram UML vs Tools UML open source

Tabel 3. Diagram UML vs Tools UML open source

DIAGRAM UML	Umletor	UML Pad	Violet	ArgoUML	Dia	StarUML	Umlet	Winbrello
	Ver 3.0.1	Ver 3.2	Ver 0.21.1	Ver 0.26.2	Ver 0.97	Ver 5.0.2.1970	Ver 9.1	Ver 5.71
1 Use-Case diagrams	✓	✓	✓	✓	✓	✓	✓	✓
2 Class diagrams	✓	✓	✓	✓	✓	✓	✓	✓
3 Object diagrams			✓	✓	✓	✓	✓	✓
4 Sequence diagrams	✓	✓	✓	✓	✓	✓	✓	✓
5 Collaboration diagrams				✓	✓	✓	✓	✓
6 Statechart diagrams	✓	✓	✓	✓	✓	✓	✓	✓
7 Activity diagrams	✓	✓	✓	✓	✓	✓	✓	✓
8 Component diagrams					✓	✓	✓	✓
9 Deployment diagrams				✓	✓	✓	✓	✓
Installation	copy	copy	copy	deb.p	deb.p	deb.p	copy	copy
Type File	File	Application	File	Application	Application	Application	Application	Application

Tabel 4. Total Predicted Time - Task Use Case Diagram

ACTIVITY	TOTAL PREDICTED TIME (in seconds)							
	Umletor	UML Pad	Violet	ArgoUML	Dia	StarUML	Umlet	Winbrello
	Ver 3.0.1	Ver 3.2	Ver 0.21.1	Ver 0.26.2	Ver 0.97	Ver 5.0.2.1970	Ver 9.1	Ver 5.71
1 Menggambar actor "Customer"	-	-	-	-	-	-	-	-
2 Memberi nama actor "Customer"	-	9.78	11.06	10.26	10.26	9.16	12.86	9.38
3 Menggambar UseCase "Withdraw cash"	-	-	-	-	-	-	-	-
4 Memberi nama UseCase "Withdraw cash"	-	10.88	10.66	11.30	10.26	10.26	15.76	10.48
5 Menggambar UseCase "Check balance"	-	-	-	-	-	-	-	-
6 Memberi nama UseCase "Check balance"	-	10.88	10.66	11.30	10.26	10.26	15.76	10.48
7 Menggambar UseCase "Print mini-statement"	-	-	-	-	-	-	-	-
8 Memberi nama UseCase "Print mini-statement"	-	12.42	12.20	12.90	11.80	11.80	17.30	12.02
9 Menggambar simbol association	-	18.60	10.60	18.60	18.60	18.60	32.40	9.20
10 Menghapus multiplicity	-	-	-	15.45	-	-	-	-
11 Menggambar simbol System Boundary	-	-	-	5.65	7.30	7.30	15.31	7.30
12 Memberi nama System Boundary	-	-	-	-	10.66	-	-	8.28
TOTAL PREDICTED TIME	-	62.56	55.18	85.58	79.14	67.38	109.39	67.14

Tabel-3 di atas menjelaskan kemampuan tools UML open source untuk menggambar diagram-diagram UML. Berdasarkan penelitian penulis perangkat lunak Dia, UMLet dan Winbrello dapat digunakan untuk menggambar *Use-Case diagrams*, *Class diagrams*, *Sequence diagrams*, *Activity diagrams*, *Component diagrams* serta *Deployment diagrams*. Dengan demikian perangkat lunak Dia, UMLet dan Winbrello akan dipakai oleh penulis untuk tahap penelitian selanjutnya.

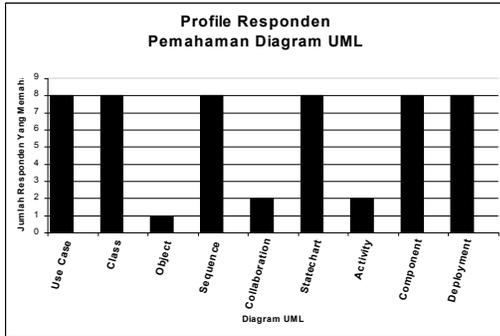
Tabel-4 di atas menjelaskan *total predicted time* perangkat lunak yang diuji dalam menggambar *Use Case Diagram*. Sesuai dengan tabel-3 berdasarkan penelitian penulis maka perangkat lunak Dia, UMLet dan Winbrello mampu untuk menggambar *Use-Case Diagrams* dengan baik dan lengkap. Dengan demikian perangkat lunak Dia, UMLet dan Winbrello akan dipakai oleh penulis untuk tahap penelitian selanjutnya.

Berdasarkan kemampuan perangkat lunak mendukung diagram-diagram UML dan *Total Predicted Time* untuk *Use Case Diagram* maka perangkat lunak Dia, UMLet dan Winbrello merupakan perangkat lunak yang akan di evaluasi lebih lanjut.

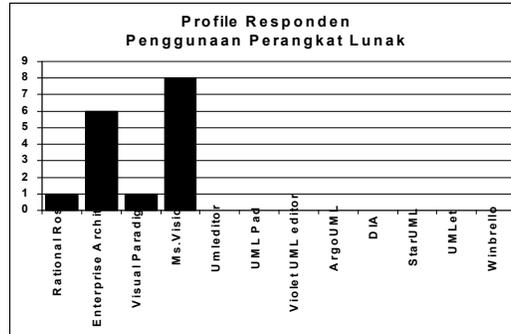
Berdasarkan tabel-5 terlihat profile responden yang sesuai dengan pemahaman responden terhadap diagram UML yang diuji coba yaitu meliputi *Use-Case Model Diagrams*, *Class diagrams*, *Sequence diagrams*, *Activity diagrams*, *Component diagrams* serta *Deployment diagrams*. Kecuali untuk *Activity diagrams*, responden paham *Statechart diagram* dibandingkan dengan *Activity diagrams*.

Berdasarkan tabel-6 terlihat profile responden yang belum pernah menggunakan perangkat lunak-perangkat lunak yang diuji yaitu perangkat lunak *open source* seperti yang terdaftar di tabel tersebut. Satu responden pernah menggunakan *Rational Rose*, enam responden kadang-kadang menggunakan *Enterprise Architect*, satu responden pernah mencoba *Visual Paradigm*, dan rata-rata responden sering menggunakan *Microsoft Visio* untuk membantu responden dalam menggambar diagram-diagram UML.

Tabel 5. Kategori Tingkat Keahlian Pengguna



Tabel 6. Penggunaan Perangkat Lunak

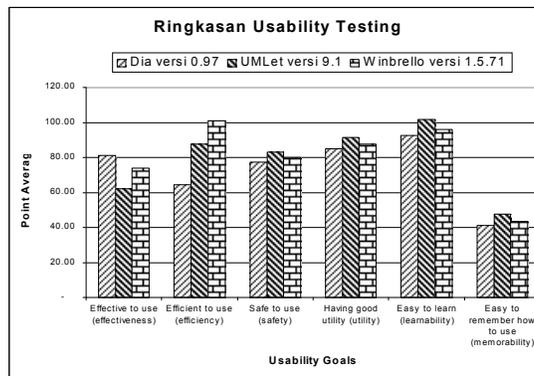


RINGKASAN USABILITY TESTING

Tabel 7. Ringkasan Usability Testing

Usability Goals	Nama Perangkat Lunak		
	Dia versi 0.97	UMLet versi 9.1	Winbrello versi 1.5.71
1 Effective to use (effectiveness)	81.13	62.13	73.75
2 Efficient to use (efficiency)	64.25	87.63	101.00
3 Safe to use (safety)	77.50	82.88	79.88
4 Having good utility (utility)	85.13	91.50	87.50
5 Easy to learn (learnability)	92.63	101.88	96.13
6 Easy to remember how to use (memorability)	41.00	47.63	43.25
Jumlah	441.64	473.65	481.51

Gambar 2. Ringkasan Usability Testing



Berdasarkan Tabel 7. Ringkasan Usability Testing, perangkat lunak yang memiliki *Effective to use* (effectiveness) dan *Efficient to use* (efficiency) yang baik adalah perangkat lunak yang memiliki nilai terkecil, yaitu UMLet dengan nilai effectiveness sebesar 62.13 dan perangkat lunak DIA dengan nilai efficiency sebesar 64.25. Perangkat lunak yang memiliki *safe to use*, *having good utility*, *easy to learn*, dan *easy to remember how to use* yang baik adalah perangkat lunak yang memiliki nilai terbesar yaitu UMLet dengan nilai safety sebesar 82.88, nilai utility sebesar 91.50, nilai learnability sebesar 101.88 dan nilai memorability sebesar 47.63. Maka berdasarkan tabel Ringkasan Usability Testing perangkat lunak UMLet dinilai memiliki usability yang baik dibandingkan perangkat lunak DIA dan Winbrello.

Perbandingan hasil Usability Testing dengan Analytical Evaluation (berdasarkan Waktu Penyelesaian Tugas) digunakan untuk mengkonfirmasi hasil Usability Testing bahwa hasil Usability Testing dapat dipakai untuk menilai Usability testing lunak yang diuji. Perbandingan kedua hasil test ini dinilai dengan melihat pola hasil test. Gambaran pola hasil test dapat dilihat pada gambar 3 untuk perangkat lunak DIA, gambar 4 untuk perangkat lunak UMLet, dan gambar 5 untuk perangkat lunak Winbrello.

Perbandingan Hasil Usability Testing dengan Analytical Evaluation (berdasarkan Waktu Penyelesaian Tugas)

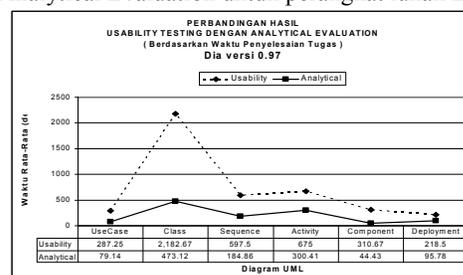
Tabel 8. Perbandingan hasil

Usability Testing dengan Analytical Evaluation

PERBANDINGAN HASIL USABILITY TESTING DENGAN ANALYTICAL EVALUATION (Berdasarkan Waktu Penyelesaian Tugas)

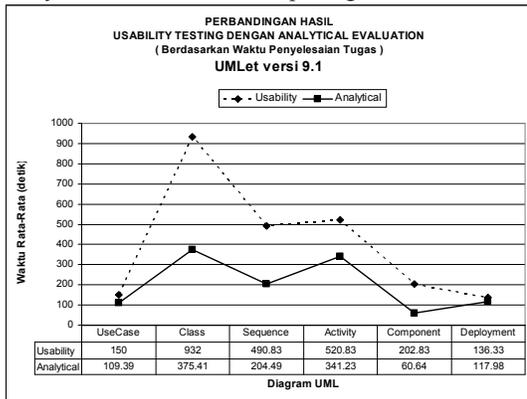
BAGIAN	Dia versi 0.97		UMLet versi 9.1		Winbrello versi 1.5.71	
	Usability Testing	Analytical Position	Usability Testing	Analytical Position	Usability Testing	Analytical Position
IV. TASK USE CASE DIAGRAM	287.25	79.14	180.00	108.39	192.00	67.14
V. TASK CLASS DIAGRAM	2102.67	473.12	932.00	375.41	1133.00	430.29
VI. TASK SEQUENCE DIAGRAM	587.00	184.85	480.83	704.48	473.83	117.77
VII. TASK ACTIVITY DIAGRAM	675.00	300.41	520.03	311.23	500.00	204.33
VIII. TASK COMPONENT DIAGRAM	310.67	44.43	202.83	80.64	115.67	35.35
IX. TASK DEPLOYMENT DIAGRAM	218.00	95.78	134.33	117.08	94.17	53.08
TOTAL BAGIAN IV s.d IX	4,271.59	1,177.74	2,432.02	1,209.14	2,560.87	955.41

Gambar 3. Perbandingan Usability Testing dng Analytical Evaluation untuk perangkat lunak DIA

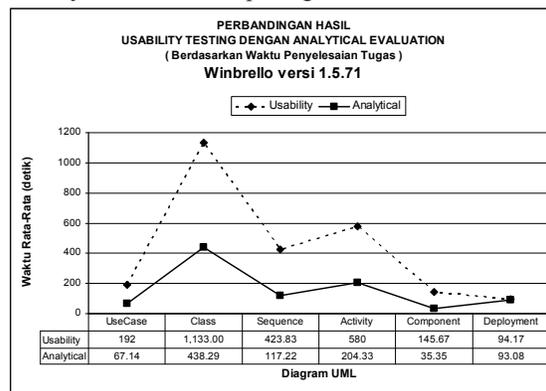


1. Diisi dengan hasil tabulasi hasil penyelesaian tugas (dalam detik)
2. Diisi dengan hasil rata-rata penyelesaian tugas (dalam detik)

Gambar 4. Perbandingan hasil Usability Testing dng Analytical Evaluation Untuk perangkat lunak UMLet



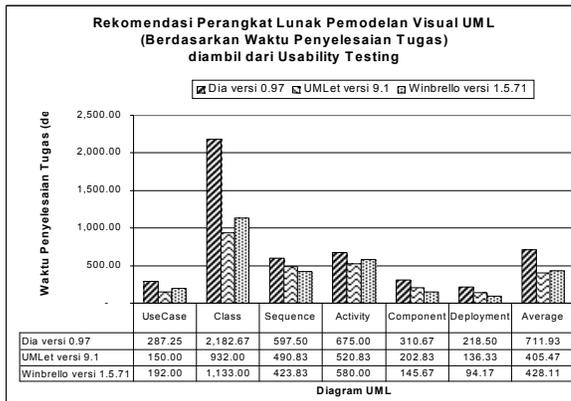
Gambar 5. Perbandingan hasil Usability Testing dng Analytical Evaluation perangkat lunak Winbrello



Berdasarkan gambar 3, 4 dan 5 yang memberikan gambaran mengenai Perbandingan hasil *Usability Testing* dengan *Analytical Evaluation* untuk masing-masing perangkat lunak Dia, UMLet dan Winbrello maka dapat ditarik kesimpulan bahwa hasil *usability testing* terhadap perangkat lunak masing-masing tersebut di atas memiliki pola yang hampir sama dengan pola yang dihasilkan oleh *analytical evaluation* untuk masing-masing perangkat lunak tersebut. Berdasarkan hal tersebut di atas maka hasil *Usability Testing* dapat dipakai untuk menentukan rekomendasi perangkat lunak yang memenuhi syarat berdasarkan *user interface* dan *usability*.

Rekomendasi Perangkat Lunak Pemodelan Visual

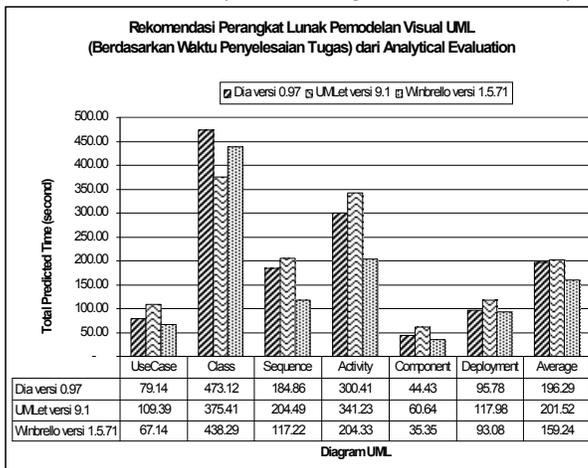
Gambar 6. Waktu Penyelesaian Tugas berdasarkan Usability Testing



Berdasarkan Tabel 7. Ringkasan Usability Testing, maka UMLet memiliki *usability* yang baik yaitu *effective to use, safe to use, having good utility, easy to learn, easy to remember how to use* walaupun untuk *efficient to use* mendapat posisi kedua dibandingkan dengan perangkat lunak DIA.

Hasil bahwa UMLet merupakan perangkat lunak dengan *usability* terbaik didukung oleh Gambar 6. Dibandingkan dengan perangkat lunak DIA, perangkat lunak UMLet memiliki waktu penyelesaian tugas yang lebih kecil untuk seluruh tugas penggambaran diagram UML. Winbrello memiliki nilai yang lebih kecil dikarenakan ada beberapa aktivitas yang tidak dapat dilakukan.

Gambar 7. Waktu Penyelesaian Tugas berdasarkan Analytical Evaluation



Berdasarkan Gambar 7 di atas perbedaan waktu rata-rata untuk menyelesaikan tugas menggambar diagram UML antara perangkat lunak UMLet dan perangkat lunak Dia hanya berbeda sedikit nilai. Untuk seluruh tugas menggambar diagram UML, dalam *analytical evaluation* perangkat lunak UMLet versi 9.1 memiliki nilai yang lebih besar dibandingkan dengan perangkat lunak DIA versi 0.97 sedangkan dalam *usability testing* perangkat lunak UMLet memiliki nilai yang lebih kecil dibandingkan dengan perangkat lunak DIA versi 0.97. Hal ini dikarenakan dalam *analytical evaluation* tidak diperhitungkan waktu yang dibutuhkan pengguna dalam usaha mencari notasi-notasi yang diperlukan dalam penggambaran diagram UML.

5. KESIMPULAN DAN SARAN

5.1. SIMPULAN

- *Functional requirement* mutlak harus dipenuhi tetapi *non-functional requirement* pun dalam hal ini sisi user interface tidak boleh diabaikan karena akan meningkat *usability* perangkat lunak tersebut.
- Pengelompokan notasi dalam *toolbox tools UML* akan membantu pengguna menggambar diagram UML.
- Perangkat lunak pemodelan visual berbasis *open source* harus dilakukan evaluasi secara berkala karena perangkat lunak ini harus dapat menyesuaikan dengan perkembangan versi UML selanjutnya.
- Dalam melakukan *usability testing* perlu dilakukan juga *analytical evaluation* untuk mengkonfirmasi hasil dari *usability testing* tersebut sehingga hasil *usability testing* dapat dipergunakan untuk evaluasi.
- Model evaluasi tools UML berbasis *open source* dari sisi user interface dan *usability* ini dapat digunakan untuk mengevaluasi perangkat lunak-perangkat lunak lainnya.
- Perangkat lunak UMLet versi 9.1 memiliki *usability* yang lebih baik dibandingkan dengan perangkat lunak DIA versi 0.97 dan perangkat lunak Winbrello versi 1.5.71

5.2. SARAN

Untuk meningkatkan penelitian selanjutnya, maka diberikan saran-saran sebagai berikut:

- Evaluasi dengan *usability testing* dapat ditingkatkan dengan melakukan *thinks aloud* dan *interviews*.
- Penelitian ini dapat ditingkatkan dengan melakukan evaluasi *CASE tools UML* berbasis *open source* mulai dari tahap pemodelan visual sampai dengan *generate source code*.

Beberapa saran untuk peningkatan perangkat lunak yang dievaluasi sehingga meningkatkan *usability* :

- Perlunya informasi (tool tips) untuk notasi di toolbox sehingga tidak membingungkan pengguna khususnya untuk perangkat lunak Winbrello.
- Perlu penambahan notasi dan penyempurnaan perangkat lunak sehingga sesuai dengan perkembangan versi UML selanjutnya (Ada beberapa notasi tidak terdapat di perangkat lunak Winbrello dan ada beberapa hal tidak dapat digambarkan dengan menggunakan perangkat lunak ini).
Contoh: notasi Interface dan Constraint tidak dapat digambarkan pada class diagram, notasi Activation tidak dapat digambarkan pada sequence diagram, kesulitan koneksi antara notasi fork, notasi join, dan notasi activity pada activity diagram, notasi interface tidak dapat digambarkan pada component diagram.
- Perlunya Kemudahan koneksi (keterhubungan antar notasi) dalam penggambaran suatu diagram UML untuk perangkat lunak Dia dan UMLet dilakukan secara manual sedangkan untuk perangkat lunak Winbrello sudah terhubung secara otomatis.
- Stabilitas perangkat lunak perlu disempurnakan (perangkat lunak Winbrello kadang-kadang menutup secara otomatis bila terjadi kesalahan sehingga semua yang telah dikerjakan menghilang dan menyulitkan user).

DAFTAR PUSTAKA

- Jeffrey L. Whitten, Lonnie D.Bentley, Kevin C.Dittman, 2004, *Metode Desain dan Analisis Sistem*, edisi 6, Penerbit Andi dan McGraw-Hill Education, Jakarta.
- Ben Shneiderman, Catherine Plaisant, 2005, *Designing the user interface: strategies for effective human-computer interaction*, 4th edition, Pearson Education, Inc, United States of America.
- Helen Sharp, Yvonne Rogers, Jenny Preece, 2006, *Interaction design: beyond human-computer interaction*, 2nd edition, John Wiley & Sons Inc, United States of America.
- Simon Bennet, John Skelton, Ken Lunn, 2005, *Schaum's Outline of UML*, second Edition, McGraw-Hill.
- Grady Booch, James Rumbough, Ivan Jacobson, 1998, *The Unified Modeling Language User Guide*, First Edition, Addison Wesley.
- Borgx, Clouseth, HarisX, IvanLanin, Kemal Prihatman, Relly Komaruzaman, 2 suntingan anonim, Sumber terbuka, 2010 [didownload 19 Maret 2010 06:48AM dari: <http://id.wikipedia.org/w/index.php?oldid=2561868>]
- Syed Mehr Ali Shah, 2009, *Usability assessment Method of the open source applications - Case Study of OpenOffice.Org 3.0* [tesis], School of Computing Blekinge Institute of Technology, Sweden.
- Silvia Abrahão¹, Emilio Iborra², Jean Vanderdonck^{1,3}, (2006), *2 Usability Evaluation of User Interfaces Generated with a Model-Driven Architecture Tool* [tesis], 1Department of Information Systems and Computation, Valencia University of Technology, Camino de Vera s/n, S-46022 Valencia, Spain. 2CARE Technologies S.A Pda. Madrigueres, 44, 03700 Denia, Alicante, Spain. 3Belgian Lab. of Computer-Human Interaction (BCHI), School of Management (IAG), Université catholique de Louvain Place des Doyens, 1 – B-1348 Louvain-la-Neuve, Belgium.